**Security Assignment #1**

Documentation

Student Name: Omar Hamed

Student Number: 1190382

Date: 1/4/2024 Tuesday

Doctor Name: Dr. Mohammed Alkhanafseh

# Introduction:

To be a part of modern world, you must have an accounts on social media platforms, those accounts of course will represent you on the virtual world, beside the banking platform accounts, so they must be secured as much as they can be, and the password that you choose to put in when you create an account is one of the most important steps that make your account secured.

But how to decide that this password is secured, it’s a good question, to any password there’s two main concepts that determine the strength of it, first is the entropy, which is how effective a password is against adversaries who try to guess it or use a **brute-force attack**.

The equation that calculate the Entropy (E) is

**E = log2(RL)**, R is the size of the pool, pool is the type of character that made the password, like if it only contains lowercase the size will be only 26 if only English used. L is the length of the password.

Second concept is variance is responsible to measures the degree of variation or spread in the ASCII values of characters in a password.

is calculated using the sum of squared differences from the mean of the ASCII values.

# Technologies:

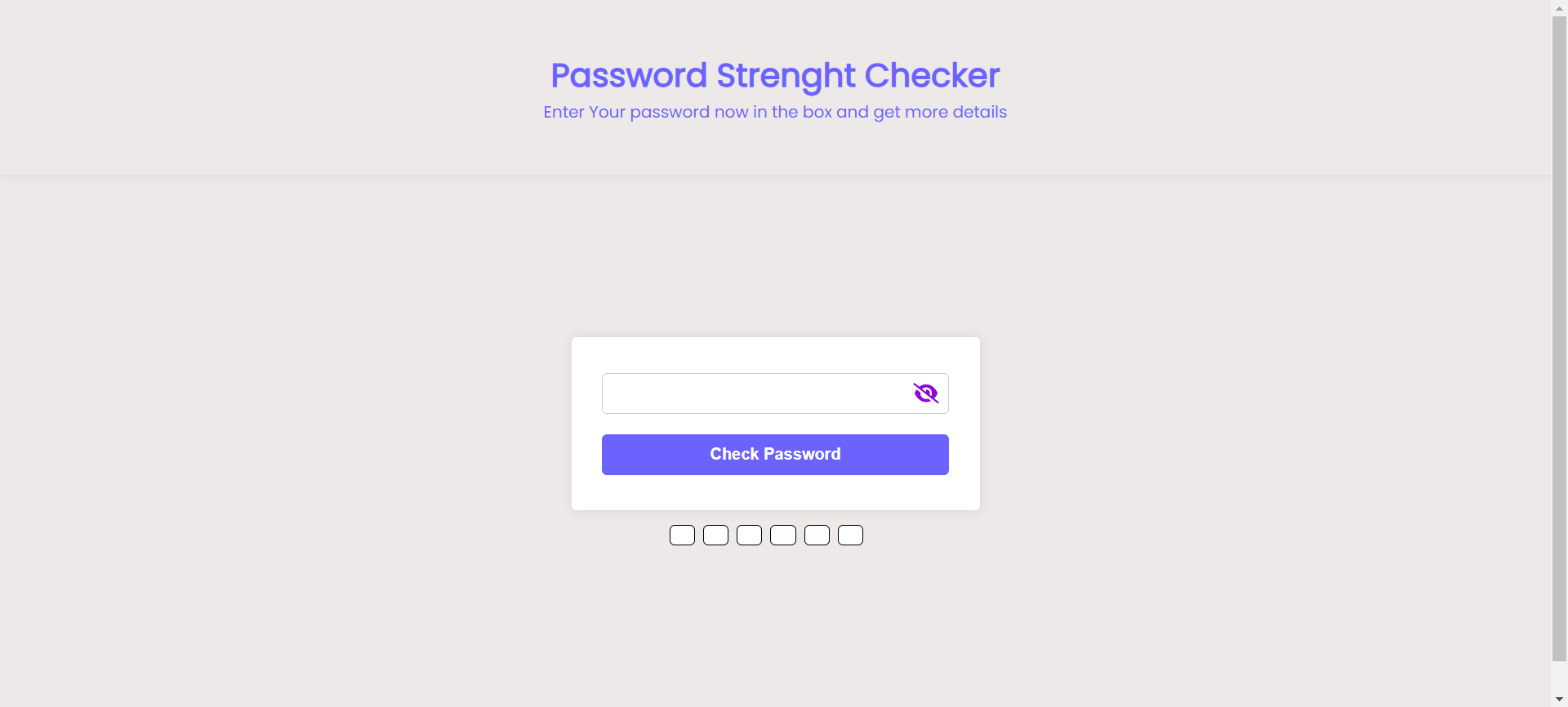
I made this program with these technologies:

1. JavaScript: for the functionality of program
2. HTML & CSS: for make the UI

# Documentation:

1. The first thing we must do is make functionality that let the user enter password and get that password:

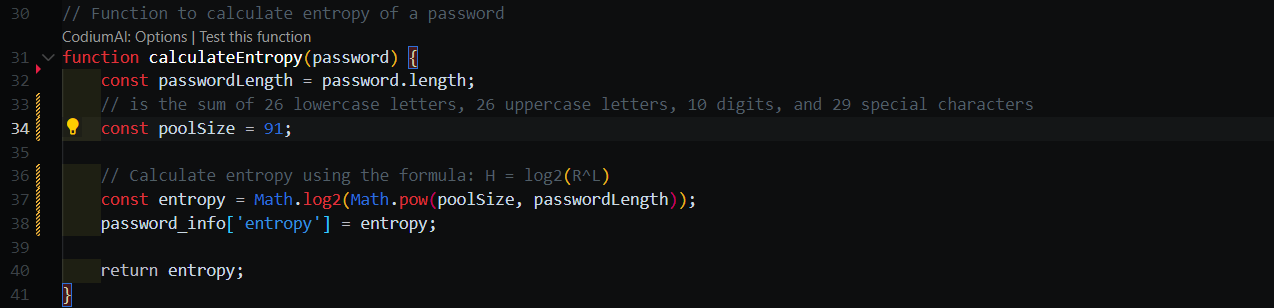
This is the main page:



I used some basic HTML and JS code to create form and get the password from the form (I will not get into the details of this process because it’s not our case study, you can take a look at code to see how it work).

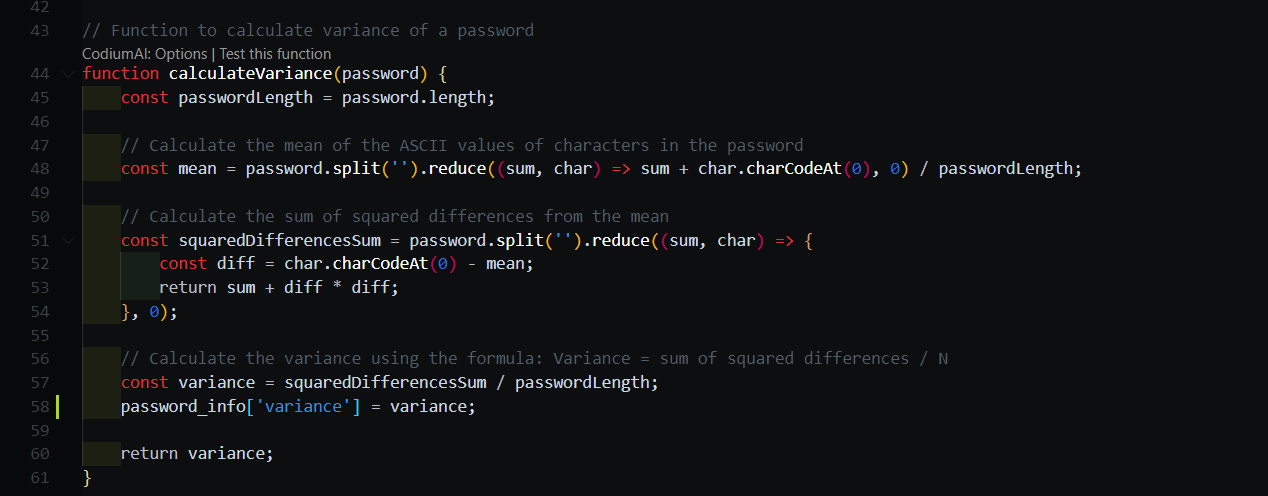
1. After the user action are handled, now it’s the important step, check password strength, these process had more than one step:

The first one is calculate the entropy and variance for the entered password:



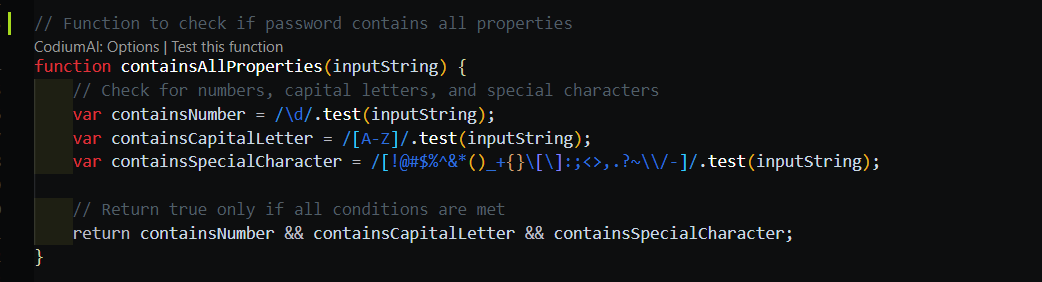
In this function we calculated the entropy be preforming the equation, first get the length of password, and decide the pool size which is 91 (uppercase 26, lowercase 26, numbers 10, special character 29), then using math library we get the entropy, save it to show it to the user and return it.

Then calculate the variance:



In this function, we first calculate the mean by get the sum for all chars ASCII code value and divide it by the length of password, then get the value of squared differences sum for the password by, get difference value by subtract the ASCII code for the character from the mean, and then add diff to the power 2 to the sum, then after finish for all character, divide it to mean to get variance.

The second step of checking the strength of password, check if it had numbers, uppercase, and special characters,

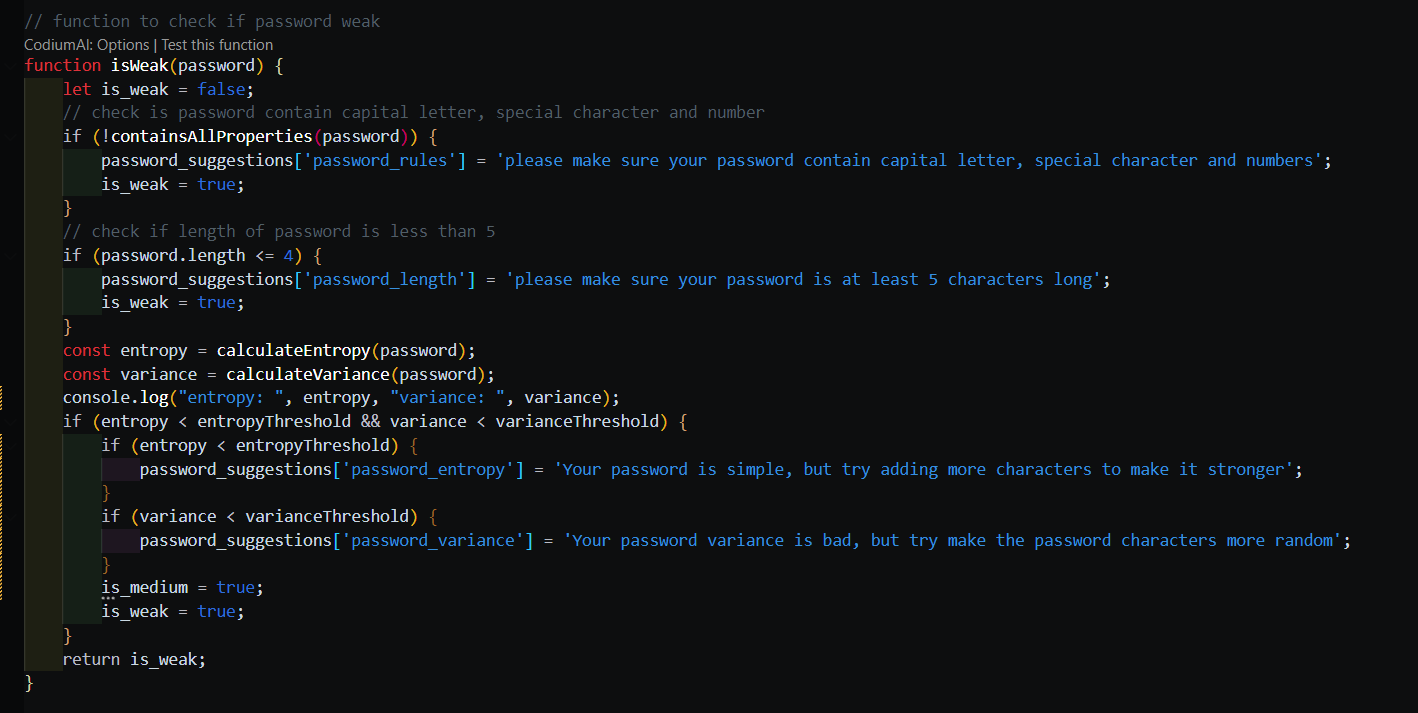


Then create a functions to decide the type of password, first one is **isWeak** which check if the password is weak based on three rules:

If had no numbers, no uppercase, and no special chars then its weak, and add to password suggestions that you must add those,

Else if the length less than 4 or equal then its weak, and you must add more characters,

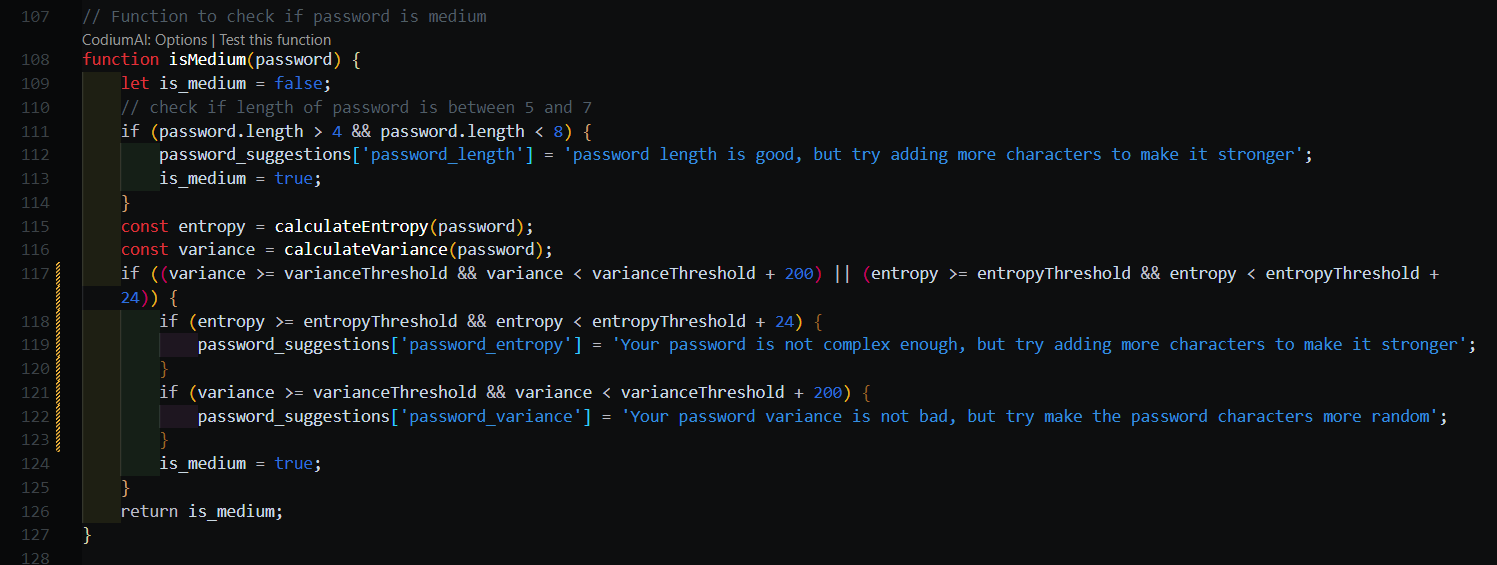
Else if entropy less than 36 and variance less than 500, then it’s not complex enough (why 36 and 500, 36 because I search it on google and found research said: less than 36 is weak, 36 < E <60 is medium, 60< E is strong, and 500 me put that rule based on so many tries)



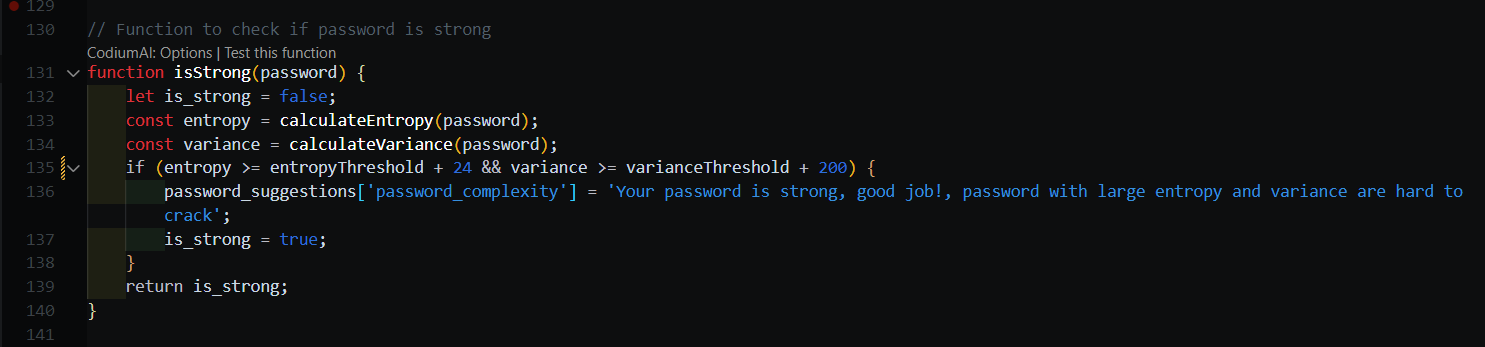
Then **isMedium** function, and rules are:

If length >4 and less than 8 its medium.

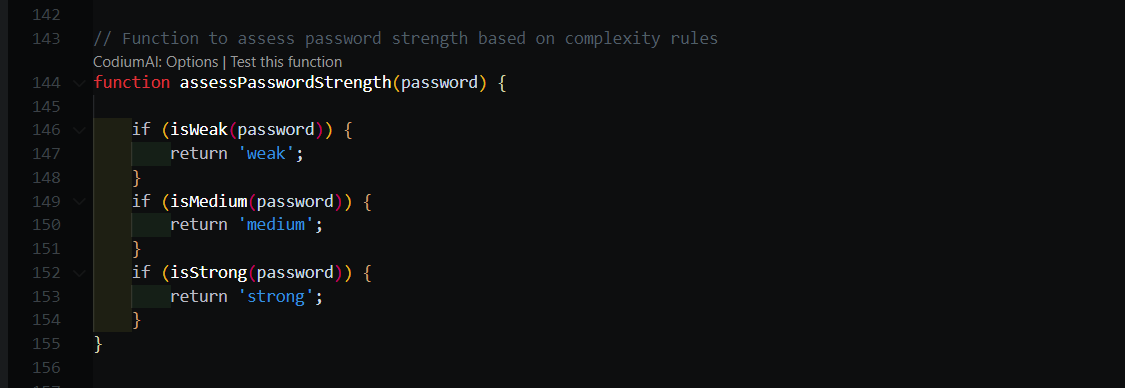
And if E >36 and E< 60 , V>500 and V < 700 then its medium .



The last thing is **isStrong** function, if the password passed the previous functions, and E>= 60 and V>=700 then it must be strong.



And function to assess the password by checking all function that we talk about,



1. The last thing is to show the result to the user, this one to show the suggestions to the user to improve the password: 

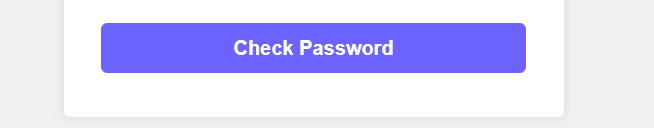
And this to update the progress bar (to display if it weak or else to the user) :



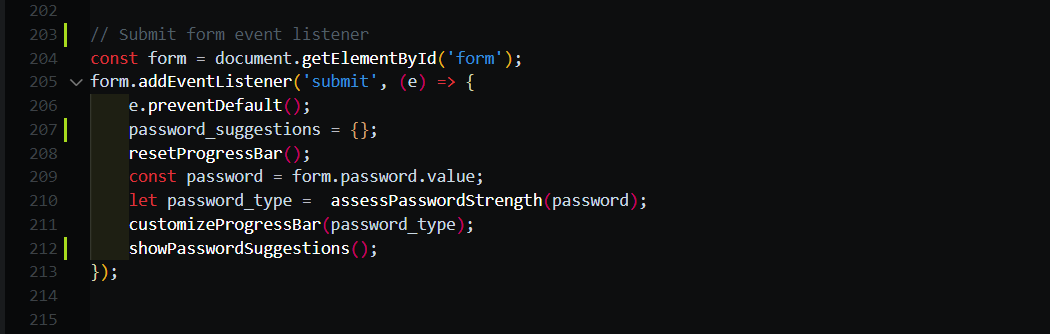
Like this:



Then the last thing that collect all the pieces together is submit function, it’s when user click in this button:



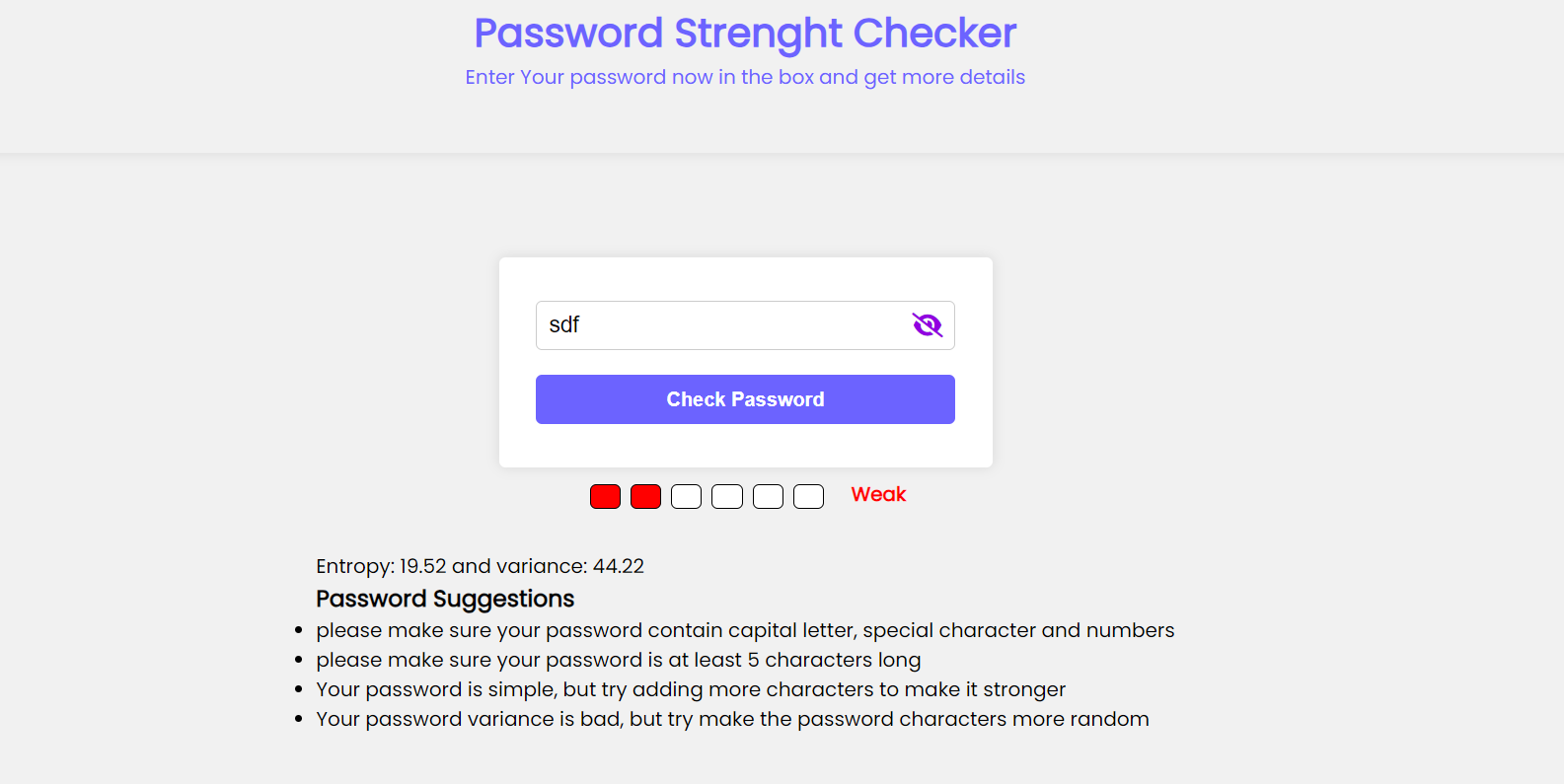
This happened:



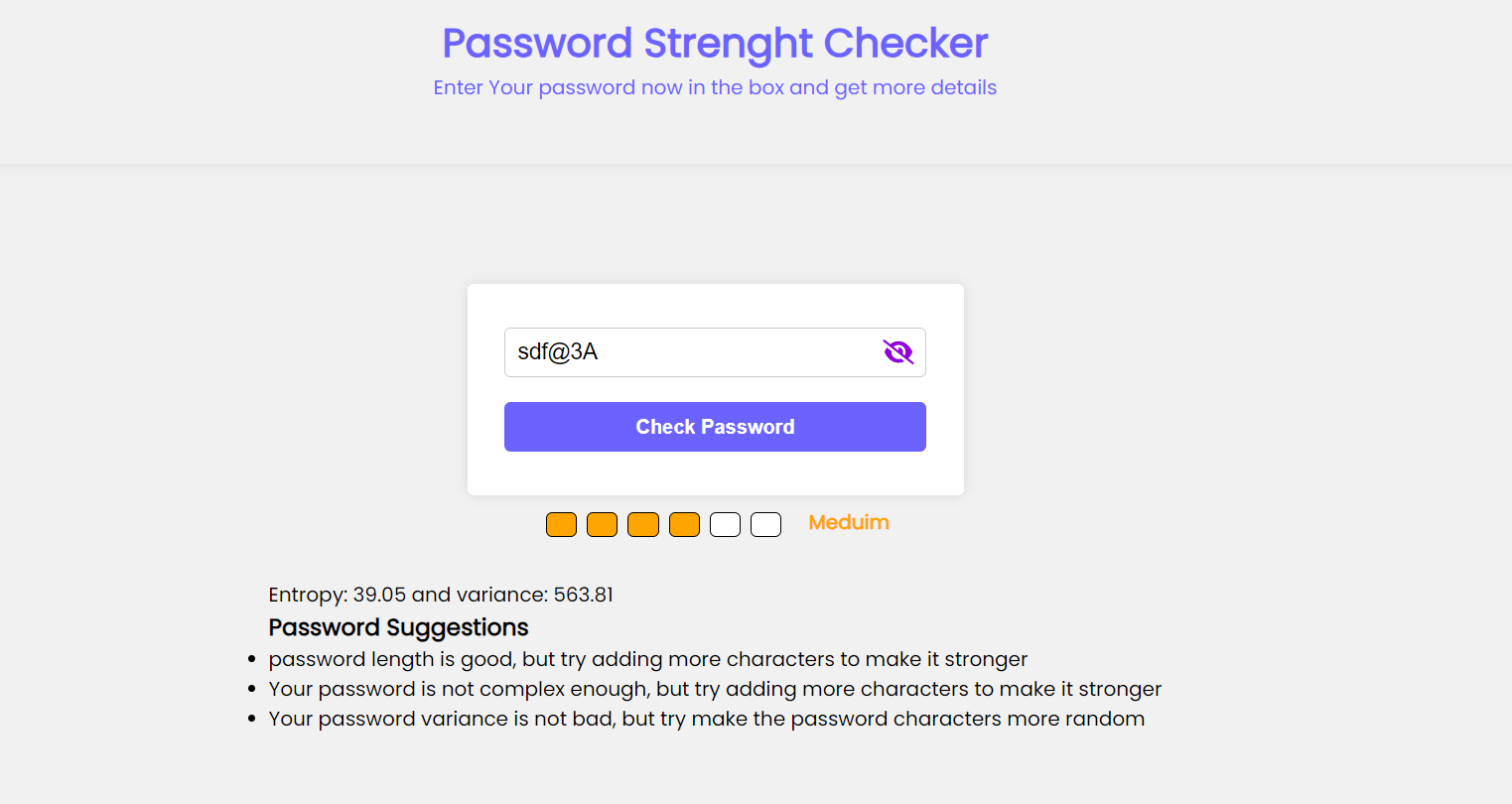
Just call all the function we talk about, and more few things that are related to browser behavior.

# Result:

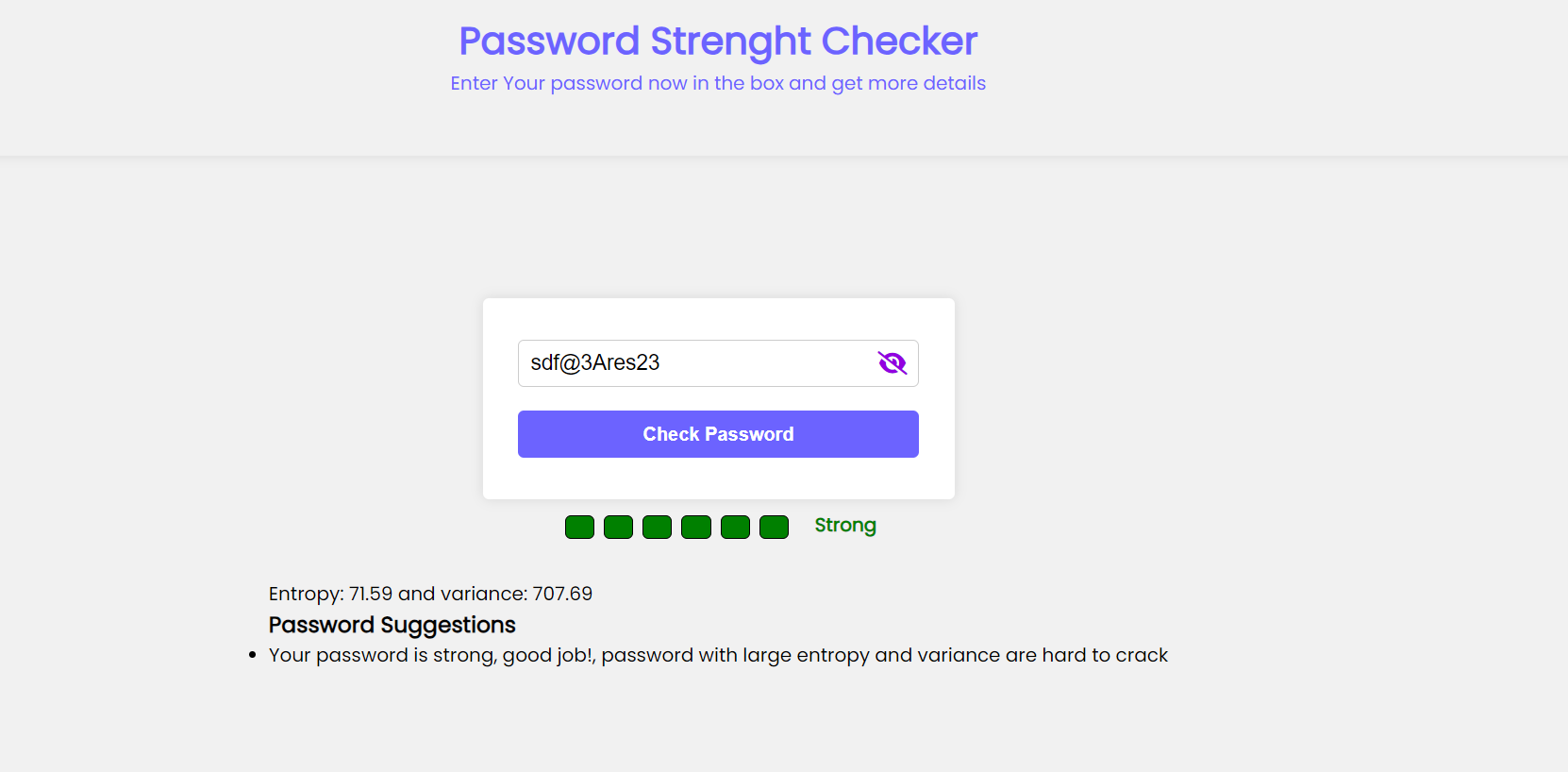
This is the result for weak password:



We can see clearly that it’s weak without any test, but for accuracy check, the suggestion can make the password stronger if we apply them. And we can also see the E and V in the screen for more accuracy.



For this password, we can see that its apply all rules, but still not enough, by add the suggestion it became stronger.



**Done**